Request for Comment
Evaluation of Fingerprint Match-on-Card Implementations
March 12, 2007

Scope

NIST seeks to conduct two evaluations of Match-on-Card (MOC) fingerprint technologies:

- An expansion of NIST's Ongoing MINEX evaluation of standardized minutiae templates to include testing of MOC implementations
- An assessment of MOC performance available from fully proprietary non-interoperable templates.

NIST seeks to evaluate implementations running on ISO/IEC 7816 cards. NIST is also interested in the more general architectures and the terms "card" and "MOC" might refer to any non-general-purpose hardware.

This document invites comment on MOC-specific testing issues. Depending on the comments received, NIST is likely to convene a workshop in due course.

Existing Testing Protocols

NIST's current MINEX evaluation is a large scale, PC based, offline assessment of the core algorithmic interoperability and accuracy of template generators and matchers using the INCITS 378 fingerprint minutia standard. The existing MINEX protocol (see MINEX API) requires suppliers to submit a software library implementing two elemental functions:

- a template generator converting a grayscale fingerprint raster to an INCITS 378 template, and
- a template matcher capable of comparing two INCITS 378 templates to produce a comparison score.

These functions are used in interoperability trials as follows:

- Supplier A's generator is invoked to produce template, TA, from an input image.
- Supplier B's generator is invoked to produce template, TB, from a different image.
- Supplier B's matching algorithm, MB, is used to compare TA and TB to produce a score = MB(TB, TA).

NIST tests proprietary-template based implementations in a separate program. This is essentially MINEX without cross-vendor matching of templates. The protocol requires suppliers to submit a software library implementing two elemental functions:

- a template generator converting a grayscale fingerprint raster to a proprietary template, and
- a template matcher capable of comparing two proprietary templates to produce comparison score.

Proposed Protocol

MOC implementations will be tested at NIST in three phases:

- Phase 1: An exact execution of the existing MINEX interoperability test and/or the existing proprietary-template test.
- Phase 2: Selected template comparisons are conducted on the smart card[1].
- Phase 3: Check that Phase 1 and Phase 2 give the same results.

NIST proposes to implement Phase 2 in a purely offline manner, using the templates produced in Phase 1. This structure imposes some unusual and *test-specific* requirements:

[1] Readers, cards and communications infrastructure would be provided to NIST by the vendor(s).

- Cards submitted for the Phase 2 test must be many times writable (to store new reference templates).
- Cards submitted for the Phase 2 test must be capable of returning a real or integer-valued similarity score, not simply a decision, to the host.
- Live users will not be used, and no fingers will be placed on a reader's sensor. Instead "match_on_card" is called. This implies the sensor subsystem must be bypassed. NIST prefers that it remain fully functional nevertheless.
- The PC-based Phase 1 algorithm must yield the same comparison scores as the Phase 2 MOC algorithm.

As described, the test requires four functionalities:

#	Return type	Function and input parameters	Executed
1	vendor_enrollment_template *	reformat_template_for_store_on_card(const INCITS_378 *reference_data);	Entirely on host PC
2	void	store_on_card(const vendor_enrollment_template *enrollment_data);	Called on PC, push to card
3	vendor_verification_template *	reformat_template_for_store_on_card(const INCITS_378 *verification_data);	Entirely on host PC
4	double	match_on_card(const vendor_verification_template *verification_data);	On card + pull from card

In any (separate) proprietary-template tests, the data structures would be different.

These functions are not supposed to be a test API. They're logical expression of needed functionality[2]. The first and third functions allow MOC implementations to:

- Transform the input INCITS 378 templates into an internal representation (prior to storing or matching). This could be the three-byte ISO/IEC 19794-2 card compact representation;
- Select a subset of the input minutiae for storage or matching on the card (for example, per the ISO/IEC 19794-2, ANNEX D guidance).

These transformations are entirely at vendor discretion, and the functions could reasonably be no-ops. The second function stores the derived data to the card. The fourth function likewise pushes a template to the card for matching.

Template generators will be called in exactly the same way as they are in the existing MINEX program. This entails conversion of a fingerprint image into an INCITS 378 template. The specifications for this template are contained in the MINEX API.

Comments

NIST solicits comments from interested parties, particularly from suppliers of smart cards and fingerprint matching algorithms, on the technical issues outlined below. Comments should be submitted to patrick.grother@nist.gov by 17:00 GMT on Thursday March 29, 2007.

NIST may publish submitted comments in a summary-of-comments document. This will be done *without attribution*. It is NIST's intention to not disclose the identity of the commenter so as to encourage candid input. Note, however, that proprietary or otherwise protected information *must **not** be sent to NIST*.

In addition to the issues identified above, NIST specifically asks for contributions on the following:

- Comments on whether the two high level functions can be implemented. What extra logical functions would be needed (for initialization, reset etc)?

[2] The calls, parameter lists and data types *will* change. Note that we have implied a "C" binding above, and would like to preserve this.

- NIST anticipates a need to use ISO/IEC 7816-4 and -11 for the testing interface. However NIST might request implementers to hide such calls behind the generic "store" and "match" functions outlined above. Is it 7816 necessary and sufficient? If so, what call sequences are needed? Are cryptographic mechanisms necessary for writing to the card (in a test)?
- What mechanisms can be used to ensure that the MOC is actually being conducted on the smart card, and not within the driver code (i.e. on the host or in the reader)? The test harness communicates with the card over an encrypted link? How would NIST write of a private key to the card? Out-of-band? Would the key be delivered to the card as a certificate?
- Whether third-party interoperable hardware/software/middleware needed?
- Whether differences between Phase 1 and Phase 2 similarity scores are inevitable. If yes, then what test should be applied to determine equivalence?
- Whether a new template-generation function call should be added to the MINEX API to also allow return of a conformant ISO/IEC 19794-2 compact card template.
- How the various operations might be timed.
- How a PC-based timing requirement might be adjusted for on-card operation.
- Any other issues.

Caution

Neither this document, nor any future execution of MOC evaluations by NIST, should be construed as an indication that NIST, nor any other agency of the US government, has decided for or against the inclusion or exclusion of the items listed below in any current or future government specification or program.

- Contactless biometric interfaces,
- MOC implementations,
- ISO/IEC 19794-2 templates, and
- Record headers in standardized templates, stored or transmitted to cards.

This document is strictly a special notice that is being published to reach a wider audience. It is not a federal procurement action, and no RFQ or RFP is available.